On the optimal compression of sets in PSPACE

Marius Zimand *

Department of Computer and Information Sciences, Towson University, Baltimore, MD, USA

Abstract. We show that if DTIME[$2^{O(n)}$] is not included in DSPACE[$2^{o(n)}$], then, for every set B in PSPACE, all strings x in B of length n can be represented by a string $compressed(x)$ of length at most $\log(|B^{=n}|) + O(\log n)$, such that a polynomial-time algorithm, given $compressed(x)$, can distinguish x from all the other strings in $B^{=n}$. Modulo the $O(\log n)$ additive trem, this achieves the information-theoretical optimum for string compression.

Keywords: compression, time-bounded Kolmogorov complexity, pseudorandom generator.

1 Introduction

In many practical and theoretical applications in computer science, it is important to represent information in a compressed way. If an application handles strings x from a finite set B, it is desirable to represent every x by another shorter string $compressed(x)$ such that $compressed(x)$ describes unambigously the initial x. Regarding the compression rate, ideally, one would like to achieve the information-theoretical bound $|compressed(x)| \leq \log(|B|)$, for all $x \in B$. If a set B is computably enumerable, a fundamental result in Kolmogorov complexity states that for all $x \in B^{=n}$, $C(x) \leq \log(|B^{=n}|) + O(\log n)$, where $C(x)$ is the Kolmogorov complexity of x, i.e., the shortest effective description of x ($B^{=n}$ is the set of strings of length n in B). The result holds because x can be described by its rank in the enumeration of $B^{=n}$. However enumeration is typically a slow operation and, in many applications, it is desirable that the unambiguous description is not merely effective, but also efficient. This leads to the idea of considering a time-bounded version of Kolmogorov complexity. An interesting line of research [Sip83,BFL01,BLvM05,LR05], which we also pursue in this paper, focuses on the time-bounded distinguishing Kolmogorov complexity, $\mathrm{CD}^{\mathrm{t}}(\cdot)$. We say that a program p distinguishes x if p accepts x and only x. $\mathrm{CD}^{\mathrm{t,A}}(\mathrm{x})$ is the size of the smallest program that distinguishes x and that runs in time $t(|x|)$ with access to the oracle A. Buhrman, Fortnow, and Laplante [BFL01] show that for some polynomial p, for every set B, and every string $x \in B^{=n}$,

* The author is supported in part by NSF grant CCF 1016158. URL: http://triton.towson.edu/~mzimand.

$\mathrm{CD}^{\mathrm{p},\mathrm{B}^{=\mathrm{n}}}(\mathrm{x}) \leq 2\log(|\mathrm{B}^{=\mathrm{n}}|) + \mathrm{O}(\log \mathrm{n})$. This is an important and very general result but the upper bound for the compressed string length is roughly $2\log(|B^{=n}|)$ instead of $\log(|B^{=n}|)$, that one may hope. In fact, Buhrman, Laplante, and Miltersen [BLM00], show that for some sets B, the factor 2 is necessary. There are some results where the upper bound is asymptotically $\log(|B^{=n}|)$ at the price of weakening other parameters. Sipser [Sip83] shows that the upper bound of $\log(|B^{=n}|)$ can be achieved if we allow the distinguisher program to use polynomial advice: For every set B, there is a string w_B of length $\mathrm{poly}(n)$ such that for every $x \in B^{=n}$, $\mathrm{CD}^{\mathrm{poly},\mathrm{B}^{=\mathrm{n}}}(\mathrm{x} \mid \mathrm{w_B}) \leq \log(|\mathrm{B}^{=\mathrm{n}}|) + \log\log(|\mathrm{B}^{=\mathrm{n}}|) + \mathrm{O}(1)$. Buhrman, Fortnow, and Laplante [BFL01] show that $\log(|B^{=n}|)$ can be achieved if we allow a few exceptions: For any B, any ϵ, for all except a fraction of ϵ strings $x \in B^{=n}$, $\mathrm{CD}^{\mathrm{poly},\mathrm{B}^{=\mathrm{n}}}(\mathrm{x}) \leq \log(|\mathrm{B}^{=\mathrm{n}}|) + \mathrm{poly}\log(\mathrm{n} \cdot 1/\epsilon)$. Buhrman, Lee, and van Melkebeek [BLvM05] show that for all B and $x \in B^{=n}$, $\mathrm{CND}^{\mathrm{poly},\mathrm{B}^{=\mathrm{n}}}(\mathrm{x}) \leq \log(|\mathrm{B}^{=\mathrm{n}}|) + \mathrm{O}((\sqrt{\log(|\mathrm{B}^{=\mathrm{n}}|)} + \log \mathrm{n})\log \mathrm{n})$, wher CND is similar to CD except that the distinguisher program is nondeterministic.

Our main result shows that under a certain reasonable hardness assumption, the upper bound of $\log(|B^{=n}|)$ holds for every set B in PSPACE.

Main Result. Assume that there exists $f \in \mathrm{E}$ that cannot be computed by circuits of size $2^{o(n)}$ with PSPACE gates. Then for any B in PSPACE, there exists a polynomial p such that for every $x \in B^{=n}$,

$$\mathrm{CD}^{\mathrm{p},\mathrm{B}^{=\mathrm{n}}}(\mathrm{x}) \leq \log(|\mathrm{B}^{=\mathrm{n}}|) + \mathrm{O}(\log \mathrm{n}).$$

The main result is a corollary of the following stronger result: Under the same hardness assumption, the distinguisher program p for x of length $\log(|B^{=n}|) + O(\log n)$ is simple conditioned by x, in the sense that $C^{\mathrm{poly}}(p \mid x) = O(\log n)$, where $C^{\mathrm{poly}}(\cdot)$ is the polynomial-time bounded Kolmogorov complexity.

We also consider some variations of the main result in which the set B is in P or in NP. We show that the hardness assumption can be somewhat weakened by replacing the PSPACE gates with Σ_3^p gates. We also show that the distinguisher program no longer needs oracle access to $B^{=n}$ in case we allow it to be nondeterministic and B is in NP.

The hardness assumption in the main result, which we call H1, states that there exists a function $f \in \mathrm{E} = \cup_{c \geq 0}\mathrm{DTIME}[2^{cn}]$ that cannot be computed by circuits of size $2^{o(n)}$ with PSPACE gates. This looks like a technical hypothesis; however, Miltersen [Mil01] shows that the more intuitive assumption "E is not contained in $\mathrm{DSPACE}[2^{o(n)}]$" implies H1. We note that this assumption (or related versions) has been used before in somewhat similar contexts. Antunes and Fortnow[AF09] use a version of H1 (with the PSPACE gates replaced by Σ_2^p gates) to show that the semi-measure $m^p(x) = 2^{-C^p(x)}$ dominates all polynomial-time samplable distributions. Trevisan and Vadhan [TV00] use a version of H1 (with the PSPACE gates replaced by Σ_5^p gates) to build for each k a polynomial-time extractor for all distributions with min-entropy $(1-\delta)n$ that are samplable by circuits of size n^k.

1.1 Discussion of technical aspects

We present the main ideas in the proof of the main result. The method is reminiscent of techniques used in the construction of Kolmogorov extractors in [FHP+06,HPV09,Zim10]. Let B in PSPACE. To simplify the argument suppose that $|B^{=n}|$ is a power of two, say $|B^{=n}| = 2^k$. If we would have a polynomial-time computable function $E : \{0,1\}^n \to \{0,1\}^k$, whose restriction on B is 1-to-1, then every $x \in B^{=n}$ could be distinguished from the other elements of $B^{=n}$ by $z = E(x)$ and we would obtain $\mathrm{CD}^{\mathrm{poly},\mathrm{B}^{=\mathrm{n}}}(\mathrm{x}) \leq |\mathrm{z}| + \mathrm{O}(1) = \log(|\mathrm{B}^{=\mathrm{n}}|) + \mathrm{O}(1)$. We do not know how to obtain such a function E, but, fortunately, we can afford a slack additive term of $O(\log n)$ and therefore we can relax the requirements for E. We can consider functions E of type $E : \{0,1\}^n \times \{0,1\}^{\log n} \to \{0,1\}^k$. More importantly, it is enough if E is computable in polynomial time given an advice string σ of length $O(\log n)$ and if every $z \in \{0,1\}^k$ has at most $O(n)$ preimages in $B \times \{0,1\}^{\log n}$. With such an E, the string $z = E(x, 0^{\log n})$ distinguishes x from strings that do not map into z and, using the general result of Buhrman, Fortnow, and Laplante [BFL01], with additional $2\log n + O(1)$ bits we can distinguish x from the other at most $O(n)$ strings that map into z. With such an E, we obtain for every $x \in B^{=n}$ the desired $\mathrm{CD}^{\mathrm{poly},\mathrm{B}^{=\mathrm{n}}}(\mathrm{x}) \leq |\mathrm{z}| + |\sigma| + 2\log \mathrm{n} + \mathrm{O}(1) = \log(|\mathrm{B}^{=\mathrm{n}}|) + \mathrm{O}(\log \mathrm{n})$.

Now it remains to build the function E. An elementary use of the probabilistic method shows that if we take $E : \{0,1\}^n \times \{0,1\}^{\log n} \to \{0,1\}^k$ at random, with high probability, every $z \in \{0,1\}^k$ has at most $7n$ preimages. The problem is that to compute a random E in polynomial-time we would need its table and the table of such a function has size $\mathrm{poly}(N)$, where $N = 2^n$. This is double exponentially larger than $O(\log n)$ which has to be the size of σ from our discussion above.

To reduce the size of advice information (that makes E computable in polynomial time) from $\mathrm{poly}(N)$ to $O(\log n)$, we derandomize the probabilistic construction in two steps.

In the first step we observe that counting (the number of preimages of z) can be done with sufficient accuracy by circuits of size $\mathrm{poly}(N)$ and constant-depth using the result of Ajtai [Ajt93]. We infer that there exists a circuit G of size $\mathrm{poly}(N)$ and constant depth such that $\{E \mid \text{every } z \text{ has} \leq 7n \text{ preimages in } B \times \{0,1\}^{\log n}\} \subseteq \{E \mid G(E) = 1\} \subseteq \{E \mid \text{every } z \text{ has} \leq 8n \text{ preimages in } B \times \{0,1\}^{\log n}\}$. Now we can utilize the Nisan-Wigderson [NW94] pseudo-random generator NW-gen against constant-depth circuits and we obtain that, for most seeds s (which we call good seeds for NW-gen), NW-gen(s) is the table of a function E where each element $z \in \{0,1\}^k$ has at most $8n$ preimages in $B \times \{0,1\}^{\log n}$. This method is inspired by the work of Musatov [Mus10], and it has also been used in [Zim11]. The seed s has size $\mathrm{poly}\log(N) = \mathrm{poly}(n)$, which is not short enough.

In the second step we use the Impagliazzo-Wigderson pseudo-random generator [IW97] as generalized by Klivans and van Melkebeek [KvM02]. We observe that checking that a seed s is good for NW-gen can be done in PSPACE, and we use the hardness assumption to infer the existence of a pseudo-random generator H such that for most seeds σ of H (which we call good seeds for H), $H(\sigma)$ is a

good seed for NW-gen. We have $|\sigma| = \log |s| = O(\log n)$ as desired. Finally, we take our function E to be the function whose table is NW-gen$(H(\sigma))$, for some good seed σ for H. It follows that, given σ, E is computable in polynomial time and that every $z \in \{0,1\}^k$ has at most $8n$ preimages in $B^{=n} \times \{0,1\}^{\log n}$.

The idea of the 2-step derandomization has been used by Antunes and Fortnow [AF09] and later by Antunes, Fortnow, Pinto and Souza [AFPS07].

2 Preliminaries

2.1 Notation and basic facts on Kolmogorov complexity

We work over the binary alphabet $\{0,1\}$. A string is an element of $\{0,1\}^*$. If x is a string, $|x|$ denotes its length; if B is a finite set, $|B|$ denotes its size. If $B \subseteq \{0,1\}^*$, then $B^{=n} = \{x \in B \mid |x| = n\}$.

The Kolmogorov complexity of a string x is the length of the shortest program that prints x. The t-time bounded Kolmogorov complexity of a string x is the length of the shortest program that prints x in at most $t(|x|)$ steps. The t-time bounded distinguishing Kolmogorov complexity of a string x is the length of the shortest program that accepts x and only x and runs in at most $t(|x|)$ steps. The formal definitions are as follows.

We fix an universal Turing machine U, which is able to simulate any other Turing machine with only a constant additive term overhead in the program length. The Kolmogorov complexity of the string x conditioned by string y, denoted $C(x \mid y)$, is the length of the shortest string p (called a program) such that $U(p,y) = x$. In case y is the empty string, we write $C(x)$.

For the time-bounded versions of Kolmogorov complexity, we fix an universal machine U, that, in addition to the above property, can also simulate any Turing machine M in time $t_M(|x|) \log t_M(|x|)$, where $t_M(|x|)$ is the running time of M on input x. For a time bound $t(\cdot)$, the t-bounded Kolmogorov complexity of x conditioned by y, denoted $C^t(x \mid y)$, is the length of the shortest program p such that $U(p,y) = x$ and $U(p,y)$ halts in at most $t(|x| + |y|)$ steps.

The t-time bounded distinguishing complexity of x conditioned by y, denoted $\mathrm{CD}^{\mathrm{t}}(\mathrm{x} \mid \mathrm{y})$ is the length of the shortest program p such that

(1) $U(p,x,y)$ accepts,

(2) $U(p,v,y)$ rejects for all $v \neq x$,

(3) $U(p,v,y)$ halts in at most $t(|v| + |y|)$ steps for all v and y.

In case y is the empty string λ, we write $\mathrm{CD}^{\mathrm{t}}(\mathrm{x})$ in place of $\mathrm{CD}^{\mathrm{t}}(\mathrm{x} \mid \lambda)$. If U is an oracle machine, we define in the similar way, $\mathrm{CD}^{\mathrm{t,A}}(\mathrm{x} \mid \mathrm{y})$ and $\mathrm{CD}^{\mathrm{t,A}}(\mathrm{x})$, by allowing U to query the oracle A.

For defining t-time bounded nondeterministic distinguishing Kolmogorov complexity, we fix U a nondeterministic universal machine, and we define $\mathrm{CND}^{\mathrm{t}}(\mathrm{x} \mid \mathrm{y})$ in the similar way.

Strings $x_1, x_2, \ldots, x_k$ can be encoded in a self-delimiting way (i.e., an encoding from which each string can be retrieved) using $|x_1| + |x_2| + \ldots + |x_k| + 2\log|x_2| + \ldots + 2\log|x_k| + O(k)$ bits. For example, x_1 and x_2 can be encoded

as $\overline{(bin(|x_2|)}01x_1x_2$, where $bin(n)$ is the binary encoding of the natural number n and, for a string $u = u_1 \dots u_m$, $\overline{u}$ is the string $u_1u_1 \dots u_mu_m$ (i.e., the string u with its bits doubled).

2.2 Distinguishing complexity for strings in an arbitrary set

As mentioned, Buhrman, Fortnow and Laplante [BFL01], have shown that for any set B and for every $x \in B^{=n}$ it holds that $\mathrm{CD}^{\mathrm{poly},\mathrm{B}^{=\mathrm{n}}}(\mathrm{x}) \leq 2\log(|\mathrm{B}^{=\mathrm{n}}|) + \mathrm{O}(\log \mathrm{n})$. We need the following more explicit statement of their result.

Theorem 1. *There exists a polynomial-time algorithm A such that for every set $B \subseteq \{0,1\}^*$, every n, every $x \in B^{=n}$, there exists a string p_x of length $|p_x| \leq 2\log(|B^{=n}|) + O(\log n)$ such that*

- $A(p_x, x) =$ *accept,*
- $A(p_x, y) =$ *reject, for every* $y \in B^{=n} - \{x\}$.

2.3 Approximate counting via polynomial-size constant-depth circuits

We will need to do counting with constant-depth polynomial-size circuits. Ajtai [Ajt93] has shown that this can be done with sufficient precision.

Theorem 2. *(***Ajtai's approximate counting with polynomial size constant-depth circuits***) There exists a uniform family of circuits $\{G_n\}_{n\in\mathbb{N}}$, of polynomial size and constant depth, such that for every n, for every $x \in \{0,1\}^n$, for every $a \in \{0,\dots,n-1\}$, and for every $\epsilon > 0$,*

- *If the number of* 1*'s in x is $\leq (1-\epsilon)a$, then $G_n(x, a, 1/\epsilon) = 1$,*
- *If the number of* 1*'s in x is $\geq (1+\epsilon)a$, then $G_n(x, a, 1/\epsilon) = 0$.*

We do not need the full strength (namely, the uniformity of G_n) of this theorem; the required level of accuracy (just $\epsilon > 0$) can be achieved by non-uniform polynomial-size circuits of depth $d = 3$ (with a much easier proof, see [Vio10]).

2.4 Pseudo-random generator fooling bounded-size constant-depth circuits

The first step in the derandomization argument in the proof of Theorem 5 is done using the Nisan-Wigderson pseudo-random generator that "fools" constant-depth circuits [NW94].

Theorem 3 (Nisan-Wigderson pseudo random generator). *For every constant d there exists a constant $\alpha > 0$ with the following property. There exists a function NW-gen$: \{0,1\}^{O(\log^{2d+6} n)} \to \{0,1\}^n$ such that for any circuit G of size 2^{n^α} and depth d,*

$$|\mathrm{Prob}_{s\in\{0,1\}^{O(\log^{2d+6} n)}}[G(\textit{NW-gen}(s)) = 1] - \mathrm{Prob}_{z\in\{0,1\}^n}[G(z) = 1]| < 1/100.$$

Moreover, there is a procedure that on inputs (n, i, s) produces the i-th bit of NW-gen(s) in time $\mathrm{poly}(\log n)$.

2.5 Hardness assumptions and pseudo-random generators

The second step in the derandomization argument in the proof of Theorem 5 uses pseudo-random generators based on the assumption that hard functions exist in $\mathrm{E} = \cup_c \mathrm{DTIME}[2^{cn}]$. Such pseudo-random generators were constructed by Nisan and Wigderson [NW94]. Impagliazzo and Wigderson [IW97] strenghten the results in [NW94] by weakening the hardness assumptions. Klivans and van Melkebeek [KvM02] show that the Impagliazzo-Wigderson results hold in relativized worlds. We use in this paper some instantiations of a general result in [KvM02].

We need the following definition. For a function $f : \{0,1\}^* \to \{0,1\}$ and an oracle A, the circuit complexity $C_f^A(n)$ of f at length n relative to A is the smallest integer t such that there exists an A oracle circuit of size t that computes f on inputs of length n.

We use the following hardness assumptions.

Assumption H1:

There exists $f \in \mathrm{E}$ such that for some $\epsilon > 0$ and for some PSPACE complete set A, $C_f^A(n) \geq 2^{\epsilon \cdot n}$.

Assumption H2:

There exists $f \in \mathrm{E}$ such that for some $\epsilon > 0$ and for some Σ_3^p complete set A, $C_f^A(n) \geq 2^{\epsilon \cdot n}$.

If H1 holds, then for some oracle A that is PSPACE complete, for every k, there exists $H : \{0,1\}^{c \log n} \to \{0,1\}^n$ computable in time $\mathrm{poly}(n)$ such that for every oracle circuit C of size n^k,

$$|\mathrm{Prob}_{\sigma \in \{0,1\}^{c \log n}}[C^A(H(\sigma)) = 1] - \mathrm{Prob}_{s \in \{0,1\}^n}[C^A(s) = 1]| < o(1).$$

If H2 holds, then for some oracle A that is Σ_3^p complete, for every k, there exists $H : \{0,1\}^{c \log n} \to \{0,1\}^n$ computable in time $\mathrm{poly}(n)$ such that for every oracle circuit C of size n^k,

$$|\mathrm{Prob}_{\sigma \in \{0,1\}^{c \log n}}[C^A(H(\sigma)) = 1] - \mathrm{Prob}_{s \in \{0,1\}^n}[C^A(s) = 1]| < o(1).$$

3 Main Result

Theorem 4. *Assuming H1, the following holds: For every set B in PSPACE, there exists a polynomial p such that for every length n, and for every string $x \in B^{=n}$, there exists a string z with the following properties:*

(1) $|z| = \lceil \log(|B^{=n}|) \rceil$,

(2) $C^p(z \mid x) = O(\log n)$,

(3) $\mathrm{CD}^{\mathrm{p},\mathrm{B}^{=\mathrm{n}}}(\mathrm{x} \mid \mathrm{z}) = \mathrm{O}(\log \mathrm{n})$.

Before proving the theorem, we note that (1) and (3) immediately imply the following theorem, which is our main result.

Theorem 5. *Assumming H1, the following holds: For every set B in PSPACE there exists a polynomial p, such that for every length n, and for every string $x \in B^{=n}$,*

$$\mathrm{CD}^{\mathrm{p},\mathrm{B}^{=\mathrm{n}}}(\mathrm{x}) \leq \log(|\mathrm{B}^{=\mathrm{n}}|) + \mathrm{O}(\log \mathrm{n}).$$

Proof. (of Theorem 4) Let us fix a set B in PSPACE and let us focus on $B^{=n}$, the set of strings of length n in B. Let $k = \lceil \log |B^{=n}| \rceil$ and let $K = 2^k$. Also, let $N = 2^n$.

Definition 1. *Let $E : \{0,1\}^n \times \{0,1\}^{\log n} \to \{0,1\}^k$. We say that E is Δ-balanced if for every $z \in \{0,1\}^m$,*

$$\left|E^{-1}(z) \ \cap \ B^{=n} \times \{0,1\}^{\log n}\right| \leq \Delta \cdot \frac{|B^{=n}| \cdot n}{K}.$$

The plan for the proof is as follows. Suppose that we have a function $E : \{0,1\}^n \times \{0,1\}^{\log n} \to \{0,1\}^k$ that is Δ-balanced, for some constant Δ.

Furthermore assume that E can be "described" by a string σ, in the sense that given σ as an advice string, E is computable in time polynomial in n.

Fix x in $B^{=n}$. and let $z = E(x, 0^{\log n})$. Clearly, the string z satisfies requirement (1). It remains to show (2) and (3).

Consider the set

$$U = \{u \in B^{=n} \mid \exists v \in \{0,1\}^{\log n}, E(u,v) = z\}.$$

Since E is Δ-balanced, the size of U is bounded by $\Delta \cdot \frac{|B^{=n}| \cdot n}{K} \leq \Delta n$.

Now observe that for some polynomial p,

$$\mathrm{CD}^{\mathrm{p},\mathrm{B}^{=\mathrm{n}}}(\mathrm{x} \mid \mathrm{z}) \leq |\sigma| + 2\log(\Delta \mathrm{n}) + \mathrm{O}(\log \mathrm{n}) + \text{self-delimitation overhead.}$$

Indeed, the following is a polynomial-time algorithm using oracle $B^{=n}$ that distinguishes x (it uses an algorithm A, promised by Theorem 1, that distinguishes x from the other strings in U, using a string p_x of length $2\log(|U^{=n}|)+O(\log n) \leq 2\log(\Delta n) + O(\log n)$).

> Input: y; the strings z, σ, p_x, defined above, are also given.
> If $y \notin B^{=n}$, then reject.
> If for all $v \in \{0,1\}^{\log n}$, $E(y,v) \neq z$, then reject.
> If $A(y, p_x) =$ reject, then reject.
> Else accept.

Clearly, the algorithm accepts input y iff $y = x$. Also, since $z = E(x, 0^{\log n})$, $C^p(z \mid x) \leq |\sigma| + O(1)$. For a further application (Theorem 14), note that the above algorithm queries the oracle $B^{=n}$ a single time.

Therefore, if we manage to achieve $\sigma = O(\log n)$, we obtain that $\mathrm{CD}^{\mathrm{p},\mathrm{B}^{=\mathrm{n}}}(\mathrm{x} \mid \mathrm{z}) \leq \mathrm{O}(\log \mathrm{n})$ and $C^p(z \mid x) \leq O(\log n)$.

Thus our goal is to produce a function $E : \{0,1\}^n \times \{0,1\}^{\log n} \to \{0,1\}^k$ that using an advice string σ of length $O(\log n)$ is computable in polynomial time and is Δ-balanced for some constant Δ. Let us call this goal $(*)$.

We implement the ideas exposed in Section 1.1 and the reader may find convenient to review that discussion.

Claim 6 . *With probability at least* 0.99*, a random function* $E : \{0,1\}^n \times \{0,1\}^{\log n} \to \{0,1\}^k$ *is* 7*-balanced.*

Proof. For fixed $x \in B$, $y \in \{0,1\}^{\log n}$, $z \in \{0,1\}^m$, if we take a random function $E : \{0,1\}^n \times \{0,1\}^{\log n} \to \{0,1\}^k$, we have that $\text{Prob}[E(x,y) = z] = 1/K$. Thus the expected number of preimages of z in the rectangle $B \times \{0,1\}^{\log n}$ is $\mu = (1/K) \cdot |B| \cdot n$. By the Chernoff's bounds,[1]

$$\text{Prob}[\text{number of preimages of } z \text{ in } B \times \{0,1\}^{\log n} > 7\mu] < e^{-(6 \ln 2)\mu}.$$

Therefore, the probability of the event "there is some $z \in \{0,1\}^k$ such that the number of z-cells in $B \times \{0,1\}^{\log n}$ is $> 7\mu$" is at most $K \cdot e^{-(6 \ln 2)\mu} < 0.01$. ∎

Claim 7 . *There exists a circuit* G *of size* $\text{poly}(N)$ *and constant depth such that for any function* $E : \{0,1\}^n \times \{0,1\}^{\log n} \to \{0,1\}^k$ *(whose table is given to* E *as the input),*

(a) If $G(E) = 1$*, then* E *is* 8*-balanced,*

(b) If E *is* 7*-balanced, then* $G(E) = 1$*.*

Proof. By Theorem 2, there is a constant-depth, $\text{poly}(N)$ size circuit that counts in an approximate sense the occurrences of a string z in $\{0,1\}^k$ in the rectangle $B \times \{0,1\}^{\log n}$. If we make a copy of this circuit for each $z \in \{0,1\}^k$ and link all these copies to an AND gate we obtain the desired circuit G.

More precisely, let x_z be the binary string of length $|B| \cdot n$, whose bits are indexed as (u,v) for $u \in B, v \in \{0,1\}^{\log n}$, and whose (u,v)-bit is 1 iff $E(u,v) = z$. By Theorem 2, there is a constant-depth, $\text{poly}(N)$ size circuit G' that behaves as follows:

- $G'(x_z) = 1$ if the number of 1's in x_z is $\leq 7\frac{|B| \cdot n}{K}$,
- $G'(x_z) = 0$ if the number of 1's in x_z is $> 8\frac{|B| \cdot n}{K}$,

If the number of 1's is between the two bounds, the circuit G' outputs either 0 or 1.

The circuit G on input a table of E, will first build the string x_z for each $z \in \{0,1\}^k$, then has a copy of G' for each z, with the z-th copy running on x_z and then connects the outputs of all the copies to an AND gate, which is the output gate of G.

∎

[1] We use the following version of the Chernoff bound. If X is a sum of independent Bernoulli random variables, and the expected value $E[X] = \mu$, then $\text{Prob}[X \geq (1+\Delta)\mu] \leq e^{-\Delta(\ln(\Delta/3))\mu}$. The standard Chernoff inequality $\text{Prob}(X \geq (1+\Delta)\mu] \leq \left(\frac{e^\Delta}{(1+\Delta)^{(1+\Delta)}}\right)^\mu$ is presented in many textbooks. It can be checked easily that $\frac{e^\Delta}{(1+\Delta)^{(1+\Delta)}} < e^{-\Delta \ln(\Delta/3)}$, which implies the above form.

Claim 8 . *If we pick at random a function* $E : \{0,1\}^n \times \{0,1\}^{\log n} \to \{0,1\}^k$, *with probability at least* 0.99, $G(E) = 1$.

Proof. This follows from Claim 6 and from Claim 7 (b). ∎

Let $\tilde{N} = N \cdot n \cdot k$. Let d be the depth of the circuit G. We denote $\tilde{n} = log^{2d+6}\tilde{N}$. Note that $\tilde{n} = \text{poly}(n)$. We consider the Nisan-Wigderson pseudo-random generator for depth d given by Theorem 3. Thus,

$$\text{NW-gen} : \{0,1\}^{\tilde{n}} \to \{0,1\}^{\tilde{N}}.$$

For any string s of length $\tilde{n}$, we view NW-gen(s) as the table of a function $E : \{0,1\}^n \times \{0,1\}^{\log n} \to \{0,1\}^k$.

Claim 9 . *If we pick at random* $s \in \tilde{n}$, *with probability of* s *at least* 0.9, *it holds that NW-gen*(s) *is* 8*-balanced.*

Proof. Since G is a circuit of constant depth and polynomial size, by Theorem 3, the probability of the event "G(NW-gen(s)) = 1" is within 0.01 of the probability of the event "$G(E) = 1$," and the second probability is at least 0.99 by Claim 8. Thus the first probability is greater than 0.9. Taking into account that if G(NW-gen(s)) = 1 then NW-gen(s) is 8-balanced, the conclusion follows. ∎

Claim 10 . *Let* $T = \{s \in \{0,1\}^{\tilde{n}} \mid$ *NW-gen*(s) *is* 8*-balanced*$\}$. *Then* T *is in PSPACE.*

Proof. We need to count for every $z \in \{0,1\}^k$, the number of z-cells in the rectangle $B^{=n} \times \{0,1\}^{\log n}$ of the table of NW-gen(s). Since B is in PSPACE and since each entry in the table of NW-gen(s) can be computed in time polynomial in $\tilde{n}$, it follows that the above operation can be done in PSPACE. ∎

Claim 11 . *Assume H1. There exists a constant* c *and a function* $H : \{0,1\}^{c \log \tilde{n}} \to \{0,1\}^{\tilde{n}}$, *computable in time* $\text{poly}(\tilde{n}) = \text{poly}(n)$, *such that if* σ *is a string chosen at random in* $\{0,1\}^{c \log \tilde{n}}$, *with probability at least* 0.8, *it holds that NW-gen*($H(\sigma)$) *is* 8*-balanced.*

Proof. Under assumption H1, there exists a pseudo-random generator $H : \{0,1\}^{c \log \tilde{n}} \to \{0,1\}^{\tilde{n}}$ such that for any set A in PSPACE,

$$|\text{Prob}_{\sigma \in \{0,1\}^{c \log \tilde{n}}}[H(\sigma) \in A] - \text{Prob}_{s \in \{0,1\}^{\tilde{n}}}[s \in A]| < 0.1.$$

Since the set T is in PSPACE, $\text{Prob}_{\sigma \in \{0,1\}^{c \log \tilde{n}}}$[NW-gen($H(\sigma)$) is 8-balanced] is within 0.1 from $\text{Prob}_{s \in \{0,1\}^{\tilde{n}}}$[NW-gen($s$) is 8-balanced]. Since the latter probability is at least 0.9, the conclusion follows.

∎

We can now finish the proof of Theorem 5.

Fix $\sigma \in \{0,1\}^{c \log \tilde{n}}$ such that NW-gen($H(\sigma)$)) is 8-balanced. Note that $|\sigma| = O(\log n)$. Given σ, each entry in the table of NW-gen(($H(\sigma)$)) can be computed in time poly(n). Thus the function $E : \{0,1\}^n \times \{0,1\}^{\log n} \to \{0,1\}^k$, whose table is NW-gen(($H(\sigma)$)), satisfies the goal ($*$). ∎

4 Variations around the main result

We analyze here the polynomial-time bounded distinguishing Kolmogorov of strings in a set B that is in P or in NP. We start with the case $B \in \mathrm{P}$. The following is the analog of Theorem 4 and its main point is that assumption H1 can be replaced by the presumably weaker assumption H2.

Theorem 12. *Assuming H2, the following holds: For every set B in* P*, there exists a polynomial p such that, for every length n, and for every string $x \in B^{=n}$, there exists a string z with the following properties:*
(1) $|z| = \lceil \log(|B^{=n}|) \rceil$,
(2) $C^p(z \mid x) = O(\log n)$,
(3) $\mathrm{CD^P(x \mid z) = O(\log n)}$.

Proof. We follow the proof of Theorem 4. First note that since $B \in \mathrm{P}$, the universal machine does not need oracle access to B. We still need to justify that assumption H1 can be replaced by the weaker assumption H2.

Assumption H1 was used in Claim 11. The point was that the set $T = \{s \mid \text{NW-gen}(s) \text{ is 8-balanced}\}$ is in PSPACE and H1 was used to infer the existence of a pseudo-random generator H that fools T. If $B \in \mathrm{P}$, we can check that NW-gen(s) is sufficiently balanced using less computational power than PSPACE. Basically we need to check that for all $z \in \{0,1\}^k$,

$$|\text{NW-gen}(s)^{-1}(z) \cap B^{=n} \times \{0,1\}^{\log n}| \leq \Delta \cdot \frac{|B^{=n}| \cdot n}{K},$$

for some constant Δ. Using Sipser's method from [Sip83], there is a Σ_2^p predicate R such that

- $R(s,z) = 1$ implies $|\text{NW-gen}(s)^{-1}(z) \cap B^{=n} \times \{0,1\}^{\log n}| \leq 16 \cdot n$,
- $R(s,z) = 0$ implies $|\text{NW-gen}(s)^{-1}(z) \cap B^{=n} \times \{0,1\}^{\log n}| \geq 64 \cdot n$.

Thus there is a set $T' \subseteq \{0,1\}^{\tilde{n}}$ in Σ_3^p such that for all $s \in T'$, NW-gen(s) is 64-balanced and T' contains all s such that NW-gen(s) is 8-balanced. Note that the second property implies that $|T'| \geq 0.99 \cdot 2^{\tilde{n}}$.

Now assumption H2 implies that there exists a pseudo-random generator $H : \{0,1\}^{c \log(\tilde{n})} \to \{0,1\}^{\tilde{n}}$ that fools T'. In particular it follows that with probability of $\sigma \in \{0,1\}^{c \log(\tilde{n})}$ at least 0.8, $H(\sigma) \in T'$ and thus NW-gen$(H(\sigma))$ is 64-balanced. The rest of the proof is identical with the proof of Theorem 5. ∎

The next result is the analog of Theorem 5 for the case when the set B is in P.

Theorem 13. *Assuming H2, the following holds: For every $B \in \mathrm{P}$, there exists a polynomial p such that for all n, and for all $x \in B^{=n}$,*

$$\mathrm{CD^{poly}(x) \leq \log(|B^{=n}|) + O(\log n)}.$$

Proof. This is an immediate consequence of (1) and (3) in Theorem 12. ∎

Next we consider the case when the set B is in NP. The main point is that the assumption H1 can be replaced by H2, and that the distinguishing program does not need access to the oracle $B^{=n}$ provided it is nondeterministic.

Theorem 14. *Assuming H2, the following holds: For every set B in* NP*, there exists a polynomial p such that for every length n, and for every string $x \in B^{=n}$, there exists a string z with the following properties:*
(1) $|z| = \lceil \log(|B^{=n}|) \rceil$,
(2) $C^p(z \mid x) = O(\log n)$,
(3) $\mathrm{CD}^{\mathrm{p},\mathrm{B}^{=\mathrm{n}}}(\mathrm{x} \mid \mathrm{z}) = \mathrm{O}(\log \mathrm{n})$.
(4) $\mathrm{CND}^{\mathrm{p}}(\mathrm{x} \mid \mathrm{z}) = \mathrm{O}(\log \mathrm{n})$.

Proof. (1), (2) and (3). We only need to show that in the proof of Theorem 4, in case $B \in$ NP, the assumption H1 can be replaced by the weaker assumption H2. This is done virtually in the same way as in the proof of Theorem 13. The predicate R also needs this time to check that certain strings are in B and this involves an additional quantifier, but that quantifier can be merged with the existing quantifiers and R remains in Σ_2^p.

(4). We need to show that, at the price of replacing CD by CND, the use of the oracle $B^{=n}$ is no longer necessary. Note that the distinguisher procedure given in the proof of Theorem 4, queries the oracle only once, and if the answer to that query is NO, then the algorithm rejects immediately. Thus, instead of making the query, a nondeterministic distinguisher can just guess a witness for the single query it makes. ∎

The following is the analog of Theorem 5 in case the set B is in NP.

Theorem 15. *Assuming H2, the following holds:*
(a) For every $B \in$ NP*, there exists a polynomial p, such that for all n, and for all $x \in B^{=n}$,*

$$\mathrm{CD}^{\mathrm{p},\mathrm{B}^{=\mathrm{n}}}(\mathrm{x}) \leq \log(|\mathrm{B}^{=\mathrm{n}}|) + \mathrm{O}(\log \mathrm{n}).$$

(b) For every $B \in$ NP*, there exists a polynomial p, such that for all n, and for all $x \in B^{=n}$,*

$$\mathrm{CND}^{\mathrm{p}}(\mathrm{x}) \leq \log(|\mathrm{B}^{=\mathrm{n}}|) + \mathrm{O}(\log \mathrm{n}).$$

Proof. Statement (a) follows from (1) and (3) in Theorem 14, and (b) follows from (1) and (4) in Theorem 14. ∎

References

[AF09] L. Antunes and L. Fortnow. Worst-case running times for average-case algorithms. In *Proceedings of the 24th Conference in Computational Complexity Conference*, pages 598–303. IEEE Computer Society Press, 2009.

[AFPS07] Luis Antunes, Lance Fortnow, Alexandre Pinto, and Andre Souto. Low-depth witnesses are easy to find. In *IEEE Conference on Computational Complexity*, pages 46–51, 2007.

[Ajt93] M. Ajtai. Approximate counting with uniform constant-depth circuits. *Advances in computational complexity*, pages 1–20, 1993.

[BFL01] Harry Buhrman, Lance Fortnow, and Sophie Laplante. Resource-bounded Kolmogorov complexity revisited. *SIAM J. Comput.*, 31(3):887–905, 2001.

[BLM00] Harry Buhrman, Sophie Laplante, and Peter Bro Miltersen. New bounds for the language compression problem. In *IEEE Conference on Computational Complexity*, pages 126–130, 2000.

[BLvM05] Harry Buhrman, Troy Lee, and Dieter van Melkebeek. Language compression and pseudorandom generators. *Computational Complexity*, 14(3):228–255, 2005.

[FHP+06] L. Fortnow, J. Hitchcock, A. Pavan, N.V. Vinodchandran, and F. Wang. Extracting Kolmogorov complexity with applications to dimension zero-one laws. In *Proceedings of the 33rd International Colloquium on Automata, Languages, and Programming*, pages 335–345, Berlin, 2006. Springer-Verlag *Lecture Notes in Computer Science #4051*.

[HPV09] John M. Hitchcock, Aduri Pavan, and N. V. Vinodchandran. Kolmogorov complexity in randomness extraction. In *FSTTCS*, pages 215–226, 2009.

[IW97] R. Impagliazzo and A. Wigderson. P = BPP if E requires exponential circuits: Derandomizing the XOR lemma. In *Proceedings of the 29th Annual ACM Symposium on the Theory of Computing (STOC '97)*, pages 220–229, New York, May 1997. Association for Computing Machinery.

[KvM02] Adam Klivans and Dieter van Melkebeek. Graph nonisomorphism has subexponential size proofs unless the polynomial-time hierarchy collapses. *SIAM J. Comput.*, 31(5):1501–1526, 2002.

[LR05] Troy Lee and Andrei E. Romashchenko. Resource bounded symmetry of information revisited. *Theor. Comput. Sci.*, 345(2-3):386–405, 2005.

[Mil01] P. B. Miltersen. Derandomizing complexity classes. In P. Pardalos, J. Reif, and J.Rolim, editors, *Handbook on Randomized Computing, Volume II*. Kluwer Academic Publishers, 2001.

[Mus10] Daniil Musatov. Improving the space-bounded version of Muchnik's conditional complexity theory via "naive" derandomization. *CoRR*, abs/1009.5108, 2010. To appear in CSR 2011.

[NW94] N. Nisan and A. Wigderson. Hardness vs. randomness. *Journal of Computer and System Sciences*, 49:149–167, 1994.

[Sip83] M. Sipser. A complexity theoretic approach to randomness. In *Proceedings of the 15th ACM Symposium on Theory of Computing*, pages 330–335, 1983.

[TV00] L. Trevisan and S. Vadhan. Extracting randomness from samplable distributions. In *Proceedings of the 41st IEEE Symposium on Foundations of Computer Science*, pages 32–42, 2000.

[Vio10] Emanuele Viola. Randomness buys depth for approximate counting. *Electronic Colloquium on Computational Complexity (ECCC)*, 17:175, 2010.

[Zim10] M. Zimand. Possibilities and impossibilities in Kolmogorov complexity extraction. *SIGACT News*, 41(4), December 2010.

[Zim11] M. Zimand. Symmetry of information and bounds on nonuniform randomness extraction via Kolmogorov complexity. In *Proceedings 26th IEEE Conference on Computational Complexity*, June 2011. To appear.